An attack on Zarankiewicz's problem through SAT solving

Jeremy Tan

March 3, 2022

Abstract

The Zarankiewicz function gives, for a chosen matrix and minor size, the maximum number of ones in a binary matrix not containing an all-one minor. Tables of this function for small arguments have been compiled, but errors are known in them. We both correct the errors and extend these tables in the case of square minors by expressing the problem of finding the value at a specific point as a series of Boolean satisfiability problems, exploiting permutation symmetries for a significant reduction in the work needed.

Certain results related to the graph packing formulation of the problem are used which give exact values at the edges of the function tables. Values in published tables lying deeper in the interior are *not* used, providing independent verification of the correct values in published tables which where almost entirely computed by hand. When the ambient matrix is also square we also give all non-isomorphic examples of matrices attaining the maximum, up to the aforementioned symmetries; it is found that most maximal matrices have some form of symmetry.

1 Introduction

The Erdős–Stone theorem [5] gives an asymptotically tight upper bound for the size of a H-free graph of a given order, where H is an arbitrary *non-bipartite* graph. Little is known in the case of bipartite H, and to that end Zarankiewicz [20] posed the following problem in 1951 (translated from the original French):

> Let R_n where $n > 3$ be an $n \times n$ square lattice. Find the smallest natural number $k_2(n)$ for which every subset of R_n of size $k_2(n)$ contains 4 points that are all the intersections of 2 rows and 2 columns. More generally, find the smallest natural number $k_j(n)$ for which every subset of R_n of size $k_j(n)$ contains j^2 points that are all the intersections of j rows and j columns.

In 1969 Guy [6] compiled tables of the natural generalisation of $k_j(n)$ where the ambient lattice and the selected sublattice need not be square, but the sublattice cannot be transposed. There is at least one error in his (hand-computed) tables, however, as discovered by Héger [7]. Merely computing values of $k_j(n)$ also does not provide a complete list of all sublattice-free maximal point sets, which may themselves have many symmetries as the Turán graphs do in their role as extremal K_n-free graphs and which may give insights as to the size and structure of further maximal examples.

This paper gives the results of a Boolean satisfiability (SAT)-based approach to Zarankiewicz's problem, motivated by its recent successes in solving very hard combinatorial problems like the fifth Schur number [8] and Keller's conjecture in seven dimensions [2]. Even though much less computational effort was spent here – all SAT solving was done on a single laptop computer – already for modestly sized cases the solution is not as trivial as a straight conversion to conjunctive normal form (CNF). The results presented here nevertheless represent a significant extension, both in the range of known values for the non-square generalisation of $k_j(n)$ and (in selected cases) a listing of all maximal examples.

1.1 Definitions and scope

Definition 1. The Zarankiewicz function $z_{a,b}(m,n)$ is the maximum number of ones in an $m \times n$ $(0,1)$-matrix with no all-one $a \times b$ minor (such matrices are called *admissible*). Indices are omitted when $a = b$ and when $m = n$, and a matrix achieving the maximum number of ones for a given set of parameters is called *maximal.*

Every $(0,1)$-matrix can be interpreted as the biadjacency matrix of a bipartite graph, so $z_a(n)$ is also the maximum size of a $K_{a,a}$-free bipartite graph whose bipartitions have n vertices each. Certain expressions are made simpler with $z_a(n)$ instead of Zarankiewicz's [20] and Guy's [6] $k_a(n) = z_a(n) + 1$, so the z-function will be used in the sequel.

In this paper only the cases $a = b = 2, 3, 4$ will be considered, while maximal matrices will always be discussed up to isomorphism of the equivalent bipartite graphs. The full set of maximal matrices will only be computed when in addition $m = n$; only the value of z is of interest otherwise, with *one* maximal matrix serving as a lower bound complemented by an upper bound proof that adding another one always leads to an all-one $a \times b$ minor.

2 Exact values, bounds and arguments

A handful of arguments are listed in Guy [6] as useful in finding specific values of $z_{a,b}(m,n)$. The three most relevant to this paper are:

Argument A. *Any admissible matrix with column sums c_i, $1 \le i \le n$, must satisfy $\sum_i \binom{c_i}{a} \le (b-1)\binom{m}{a}$. Otherwise, by the pigeonhole principle – where pigeons are a-subsets of ones in each column, each such subset potentially part of an all-one $a \times b$ minor, and holes are all a-subsets of the matrix's rows – there is a hole with at least b pigeons, forming an all-one $a \times b$ minor.*

Argument B. *For non-negative integers m, n, k with $m - n > 1$ and $k \ge 2$, $\binom{m-1}{k} + \binom{n+1}{k} < \binom{m}{k} + \binom{n}{k}$. Hence the binomial sum over columns $\sum_i \binom{c_i}{a}$ in argument A is minimised by distributing ones so that no two column sums differ by more than 1; if the binomial sum is then equal to $(b-1)\binom{m}{a}$ and the matrix can still be made to have no all-one $a \times b$ minor, that matrix must be maximal.*

Corollary 2.1 (Čulik [4])**.** *If $1 \le a \le m$ and $n \ge (b-1)\binom{m}{a}$, $z_{a,b}(m,n) = (a-1)n + (b-1)\binom{m}{a}$.*

Argument D. *Take any row of any admissible matrix. If this row's ones lie in columns with sums $c_1, \ldots, c_r$, the inequality $\sum_{i=1}^{r} \binom{c_i-1}{a-1} \le (b-1)\binom{m-1}{a-1}$ must hold, for otherwise (by argument A) there is an all-one $(a-1)\times b$ minor extendable to an all-one $a \times b$ minor through the ones in the chosen row.*

The above arguments all have a "transposed" form obtained by replacing "columns" with "rows" and vice versa. The following inclusion argument is also clear.

Argument I. *For $m' \le m$ and $n' \le n$, every $m' \times n'$ minor of every witness to $z_{a,b}(m,n)$ is admissible and thus has at most $z_{a,b}(m',n')$ ones.*

A useful explicit upper bound, with equality in a wider range of cases than that provided by Čulik's theorem [4], is given by the following.

Theorem 2.2 (Roman [12])**.** *For all integers $p \ge a-1$*

$$z_{a,b}(m,n) \le \frac{b-1}{\binom{p}{a-1}}\binom{m}{a} + \frac{(p+1)(a-1)}{a}n$$

and equality holds with $p = a$ or $p = a-1$ when $(b-1)\binom{m}{a} - aT(m,a,b) \le n$, where $T(m,a,b)$ is the largest number of complete a-uniform complete hypergraphs on $a+1$ vertices (K^a_{a+1}) that can be packed into a $b-1$-fold K^a_m. The lower bound, which is approximately $\frac{b-1}{a+1}\binom{m}{a}$, may be reduced by $a-1$ if the packing is not exact.

This bound appears tighter or at least as tight as other general bounds in the literature, such as the one developed by Collins [3], so it is the bound given in the tables in section 4.

3 Method

Beyond the range of arguments for which theorem 2.2 gives a proven exact value for the z-function, the SAT-based approach calls for encoding an instance of the problem with a, b, m, n and a guess w for the corresponding z into one or more CNFs, conjunctions (AND) of clauses or disjunctions (OR) of Boolean variables. The basic encoding is very simple: one variable for each entry of the $m \times n$ $(0,1)$-matrix A, one clause for each and every $a \times b$ minor in rows $r_1, \ldots, r_a$ and columns $c_1, \ldots, c_b$

$$\bigvee_{i=1}^{a} \bigvee_{j=1}^{b} \neg a_{ij}$$

and a cardinality constraint requiring A to have exactly w ones (its encoding details are discussed below). Any solution to this CNF forms an admissible matrix, proving $z \ge w$; conversely if the instance is unsatisfiable (UNSAT) this indicates $z < w$. Most SAT solvers have an option to output a concrete, machine-verifiable UNSAT proof if the instance turns out that way [17].

To this basic scheme we add some major optimisations, without which extending the range of known Zarankiewicz function values would not be possible.

Algorithm 1 Admissible (by arguments A and I) column partition generator

```
p ← empty stack                              ▷ workspace for building up partitions in descending order
procedure P(a, b, m, n, w)
    L_A ← (b − 1) C(m, a)                    ▷ only set at procedure start, immutable afterwards
    if ∑p > L_A or p is inadmissible by argument I then
        return
    else if w = 0 then
        yield p
    else if k > (m − 1)n then                ▷ by the pigeonhole principle, some further columns must sum to m
        d ← k − (m − 1)n
        push m d times onto p
        P(a, b, m, n − d, w − dm)
        pop d times from p
    else
        for t ∈ [⌈w/n⌉, min(w, m)] do        ▷ all possible values for the next part
            push t onto p
            P(a, b, t, n − 1, w − t)
            pop from p
        end for
    end if
end procedure
```

Here $L_A \leftarrow (b-1)\binom{m}{a}$.

3.1 Generating partitions

An admissible or maximal matrix clearly remains as such under all row and column permutations. It is therefore enough for a given w to solve instances where the row and column sums are *fixed*, over all possible combinations of *unordered* row and column partitions not forbidden by the arguments of section 2 – an approach very much like Heule's cube-and-conquer paradigm [9]. To generate all such partitions efficiently we use algorithm 1.

"Inadmissible by argument I" means that for some n' the sum of the n' deepest elements of the current p is already strictly greater than $z_{a,b}(m, n')$, forcing an all-one $a \times b$ minor to appear in these first n' columns. The algorithm to generate row partitions is similar.

Once all possible row and column partitions have been obtained argument D can then be used to remove partition pairs (considering the row with the most, r, ones and the r columns with the *least* ones – if argument D fails for this most pessimal column choice it must also fail for all other column choices – and vice versa).

3.2 Cardinality constraints

To express that exactly k out of n bits $b_1, \ldots, b_n$ should be true we use the equality variant of Sinz's sequential counter encoding [14] as described, tested and deemed fastest for general use among different cardinality constraint encodings by Wynn [18]. $k(n-k)$ auxiliary variables $a_{i,j}$ are used where $1 \le i \le k$ and $1 \le j \le n-k$, with the following clauses (all literals $a_{i,j}$ with i or j outside their specified ranges are dropped):

$$\bigwedge_{i=1}^{k} \bigwedge_{j=1}^{n-k-1} \neg a_{i,j} \vee a_{i,j+1}$$

$$\bigwedge_{i=0}^{k} \bigwedge_{j=1}^{n-k} \neg a_{i,j} \vee a_{i+1,j} \vee \neg b_{i+j}$$

$$\bigwedge_{i=1}^{k-1} \bigwedge_{j=1}^{n-k} a_{i,j} \vee \neg a_{i+1,j}$$

$$\bigwedge_{i=1}^{k} \bigwedge_{j=0}^{n-k} a_{i,j} \vee \neg a_{i,j+1} \vee b_{i+j}$$

This encoding has two desirable properties:

- If a partial assignment of the b_i is such that said assignment cannot be completed without violating the cardinality constraint, unit propagation alone will lead to a contradiction (empty clause).
- If exactly k of the b_i are assigned true, unit propagation alone will assign the other b_i false.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{sort rows}} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{sort columns}} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 1: Reverse-lexicographically sorting a $(0,1)$-matrix to a fixed point.

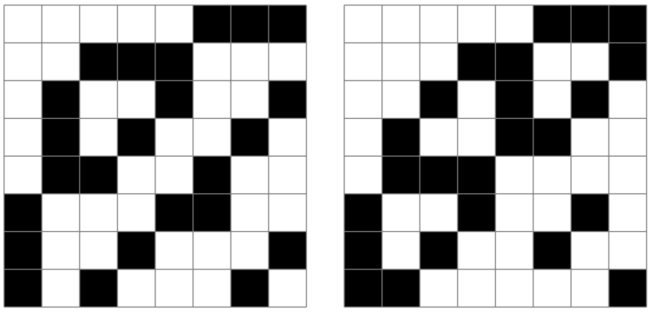


Figure 2: Two non-identical yet isomorphic maximal matrices (for $a = b = 2$, $m = n = 8$) that satisfy all constraints in section 3.

Since unit propagation is hardwired into all state-of-the-art SAT solvers, using the above encoding should result in faster rejection of partially filled matrices that cannot be completed to an admissible matrix.

3.3 Lexicographic constraints

Even with fixed row and column sums, there still remain the symmetries of swapping two rows or two columns with the *same* sum. These symmetries are broken by requiring groups of rows or columns with the same sum to be contiguous and lexicographically sorted; every $(0,1)$-matrix can be permuted to satisfy this property by the following theorem.

Theorem 3.1. *Lexicographically sorting rows and columns of any $(0,1)$-matrix A alternately as in Figure 1 will reach a fixed point (both rows and columns sorted) in a finite number of steps. This remains true even if the sets of rows and columns are partitioned so that rows and columns cannot move across partitions.*

Proof. With rows and columns indexed starting from 0, define $f(A) = \sum_i \sum_j 2^{i+j} a_{ij}$. Swapping rows/columns a and b where $a < b$ but the numerical value n_a of column a is greater than n_b changes $f(A)$ by $2^b n_a + 2^a n_b - 2^a n_a - 2^b n_b = (2^b - 2^a)(n_a - n_b) > 0$, i.e. sorting two out-of-order rows/columns strictly increases (or decreases, if sorting in reverse order) $f(A)$, which is clearly integral and bounded by 0 from below and $\sum_i \sum_j 2^{i+j}$ from above. Since there are a finite number of possibilities for each value in the strictly monotone sequence of $f(A)$'s generated, it must terminate at a point when A is sorted *both* in rows and columns. $\square$

Given two equal-length strings of Boolean variables $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, the binary number represented by the a_i may be constrained to be *at most* that represented by the b_i (where a_1, b_1 are most significant) through $n-1$ auxiliary variables $c_1, \ldots, c_{n-1}$ and the clauses (c_0 and c_n are dropped)

$$\bigwedge_{i=1}^{n-2} \neg c_i \vee c_{i+1}$$

$$\bigwedge_{i=1}^{n} c_{i-1} \vee \neg a_i \vee b_i \qquad \bigwedge_{i=1}^{n} c_{i-1} \vee a_i \vee b_i \vee \neg c_i$$

$$\bigwedge_{i=1}^{n} c_{i-1} \vee \neg a_i \vee \neg b_i \vee \neg c_i \qquad \bigwedge_{i=1}^{n} c_{i-1} \vee a_i \vee \neg b_i \vee c_i$$

In our application of this form of symmetry breaking to the problem at hand the sort order is reversed: 1 comes before 0. The cardinality and lexicographic constraints do not remove all symmetries of a $(0,1)$-matrix (see figure 2) – doing so would require solving the graph isomorphism problem – but they are nevertheless very useful in reducing the number of instance solutions.

3.4 Software

All SAT solving was done with Kissat [1] on one laptop computer with the `--sat` and `--unsat` flags set according to whether or not a solution was expected, and no other settings touched. The maximal matrices in the $m = n$ case were filtered to remove isomorphs using the `shortg` utility in nauty [10]; the automorphism groups of the corresponding bipartite graphs were computed using GAP (`https://gap-system.org`).

The partitioning and CNF-building code written for this project, together with the raw results obtained, is available in our Kyoto repository [15].

4 Tables for the Zarankiewicz function

The following three tables are corrected and extended versions of the tables for $z_a(m, n)$ given in Guy [6] where $a = 2, 3, 4$. Values above solid lines are both exact and given by theorem 2.2; the dashed lines indicate the limits of Guy's tables and grey backgrounds indicate errors Guy made. A bold value is exact, proven by the methods in this paper; other values are the upper bounds given by theorem 2.2.

4.1 Discussion

The last section of Héger's thesis [7] is devoted to proving exact values and tighter bounds for $z_2(m, n)$, which is closely related to finite geometries by Reiman's construction [11]: the point-line incidence matrix of the projective plane of prime power order q furnishes a maximal matrix for $z_2(q^2 + q + 1)$, showing that it is equal to $(q + 1)(q^2 + q + 1)$. Héger collects the new results into another table for $z_2(m, n)$, which has some values marked exact that are not marked as such in table 1, but that table comes with a caveat:

> In some cases we did rely on the exact values reported by Guy. Possibly undiscovered inaccuracies there may result in inaccurate values here as well.

By computing the z-values through an independent method we have completed the list of errors in Guy's $z_a(m, n)$ tables – there are only eight such errors, all in the $z_2(m, n)$ table, and all are too low by just one. There are meanwhile no discrepancies in the values marked as exact in both Héger's table and table 1, where Héger did not rely on Guy.

The link to finite geometries does not carry over to larger minor sizes, where the bound of theorem 2.2 appears to be less sharp, particularly when $m \approx n$. Better bounds at the edges of the exact region can often be derived by applying the arguments of section 2 to eliminate all possible partitions and hence the need for any SAT solving; this gives for example $z_4(11, 14) \leq 106$.

$m \setminus n$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**
3		**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**
4			**9**	**10**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**
5				**12**	**14**	**15**	**17**	**18**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**
6					**16**	**18**	**19**	**21**	**22**	**24**	**25**	**27**	**28**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
7						**21**	**22**	**24**	**25**	**27**	**28**	**30**	**31**	**33**	**34**	**36**	**37**	**39**	**40**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**
8							**24**	**26**	**28**	**30**	**32**	**33**	**35**	**36**	**38**	**39**	**41**	**42**	**44**	**45**	**47**	**48**	**50**	**51**	**53**	**54**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**
9								**29**	**31**	**33**	**36**	**37**	**39**	**40**	**42**	**43**	**45**	**46**	**48**	**49**	**51**	**52**	**54**	**55**	**57**	**58**	**60**	**61**	**63**	**64**	**66**	**67**	**69**	**70**
10									**34**	**36**	**39**	**40**	**42**	**44**	**46**	**47**	**49**	**51**	**52**	**54**	**55**	**57**	**58**	**60**	**61**	**63**	**64**	**66**	**67**	**69**	**70**	**72**	**73**	**75**
11										**39**	**42**	**44**	**45**	**47**	**50**	**51**	**53**	**55**	**57**	**59**	**60**	**62**	**63**	**65**	**66**	**68**	**69**	**71**	**72**	**74**	**75**	**77**	**78**	**80**
12											**45**	**48**	**49**	**51**	**53**	**55**	**57**	**60**	**61**	**63**	**65**	**66**	**68**	**70**	**72**	**73**	**75**	**76**	**78**	**79**	**81**	**82**	**84**	**85**
13												**52**	**53**	**55**	**57**	**59**	**61**	**64**	**66**	**67**	**69**	**71**	**73**	**75**	**78**	**79**	**81**	**82**	**84**	**85**	**87**	**88**	**90**	**91**
14													**56**	**58**	**60**	**63**	**65**	**68**	**70**	**72**	**73**	**75**	**78**	**80**	**82**	**84**	**86**	**87**	**89**	**91**	**92**	**94**	**96**	**98**
15														**61**	**64**	**67**	**69**	**72**	**75**	**77**	**78**	**80**	**82**	**85**	**86**	**88**	**91**	**93**	**95**	**96**	**98**	**100**	**102**	**105**
16															**67**	**70**	**73**	**76**	**80**	**81**	**83**	**85**	**87**	**90**	**91**	**93**	**96**	**98**	**100**	**102**	**103**	106	108	110
17																**74**	**77**	**80**	**84**	**85**	**87**	**89**	**91**	**94**	**96**	**98**	**101**	**102**	105	107	109	111	113	115
18																	**81**	**84**	**88**	**90**	**91**	**93**	**96**	**99**	**101**	**103**	107	109	111	113	115	117	119	121
19																		**88**	**92**	**95**	**96**	**98**	**100**	**103**	**106**	110	112	115	117	119	121	123	125	127
20																			**96**	**100**	**101**	**103**	**105**	**108**	**111**	115	117	120	122	125	127	129	131	133
21																				**105**	**106**	**108**	**110**	115	117	120	122	125	127	130	132	135	137	140
22																					**108**	**110**	**114**	120	122	125	127	130	132	135	137	140	142	145
23																						**115**	**118**	125	128	130	133	135	138	140	143	145	148	150
24																							**122**	130	133	136	139	141	144	146	149	151	154	156

Table 1: $z_2(m, n)$

$m \setminus n$	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
3	**8**	**10**	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**	**34**	**36**	**38**	**40**	**42**	**44**	**46**	**48**
4		**13**	**16**	**18**	**21**	**24**	**26**	**28**	**30**	**32**	**34**	**36**	**38**	**40**	**42**	**44**	**46**	**48**	**50**	**52**	**54**
5			**20**	**22**	**25**	**28**	**30**	**33**	**36**	**38**	**41**	**44**	**46**	**49**	**52**	**54**	**57**	**60**	**62**	**64**	**66**
6				**26**	**29**	**32**	**36**	**39**	**42**	**45**	**48**	**50**	**53**	**56**	**58**	**61**	**64**	**66**	**69**	**72**	**74**
7					**33**	**37**	**40**	**44**	**47**	**50**	**53**	**56**	**60**	**63**	**66**	**69**	**72**	**75**	**78**	**81**	**84**
8						**42**	**45**	**50**	**53**	**57**	**60**	**64**	**67**	**70**	**74**	**77**	82	85	88	92	95
9							**49**	**54**	**59**	**64**	**67**	**70**	**73**	**77**	**81**	**85**	91	94	98	101	104
10								**60**	**64**	**68**	**73**	**77**	**81**	**85**	**90**	**94**	100	104	108	112	116
11									**69**	**74**	**80**	**84**	**88**	**92**	**96**	**101**	109	113	117	121	125
12										**80**	**86**	**91**	**96**	**99**	108	113	118	122	127	132	136
13											**92**	**98**	**104**	**107**	117	122	126	131	136	140	145
14												**105**	**112**	**115**	125	130	136	141	146	151	155
15													**120**	**123**	134	139	144	150	155	160	166
16														**128**	142	148	154	160	165	170	176

Table 2: $z_3(m, n)$

$m \setminus n$	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
4	**15**	**18**	**21**	**24**	**27**	**30**	**33**	**36**	**39**	**42**	**45**	**48**	**51**	**54**	**57**	**60**	**63**	**66**
5		**22**	**26**	**30**	**33**	**37**	**41**	**45**	**48**	**52**	**56**	**60**	**63**	**66**	**69**	**72**	**75**	**78**
6			**31**	**36**	**39**	**43**	**47**	**51**	**55**	**59**	**63**	**67**	**71**	**75**	**78**	**82**	**86**	**90**
7				**42**	**45**	**49**	**54**	**58**	**63**	**68**	**72**	**77**	**82**	**87**	**90**	**95**	**100**	**105**
8					**51**	**55**	**60**	**65**	**70**	**75**	**80**	88	93	97	102	106	111	115
9						**61**	**67**	**72**	**78**	**84**	**88**	97	102	108	113	118	123	129
10							**74**	**79**	**86**	**93**	**97**	108	114	120	126	131	136	141
11								**86**	**93**	**100**	112	118	124	130	136	142	148	154
12									**100**	**108**	121	127	134	141	148	154	161	168
13										**117**	130	138	145	153	159	166	173	180

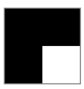
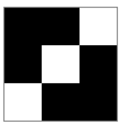
Table 3: $z_4(m, n)$

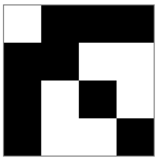
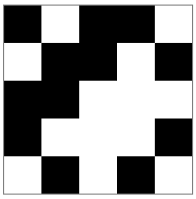
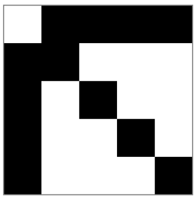
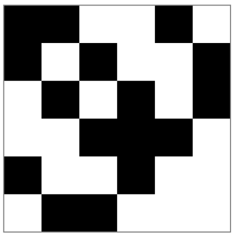
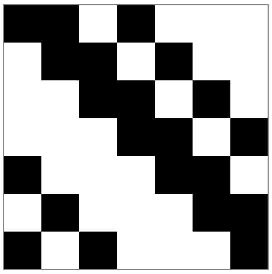
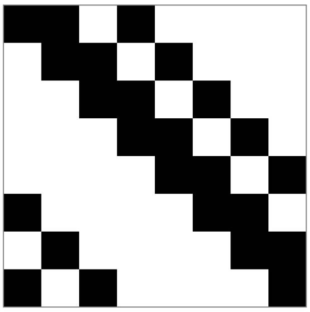
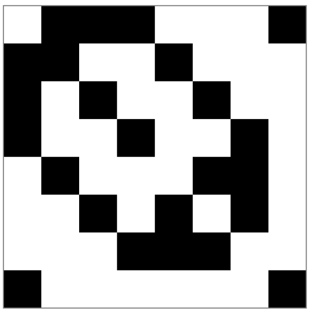
5 Maximal square matrices

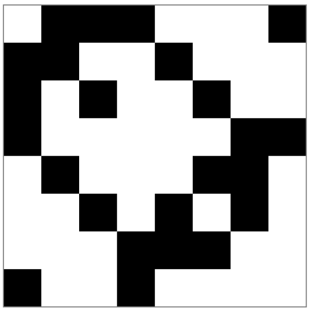
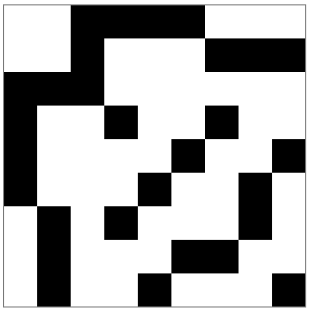
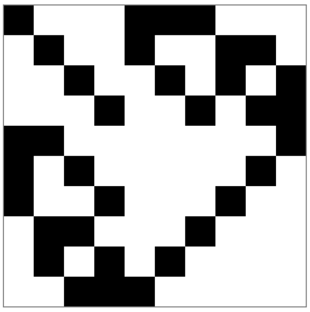
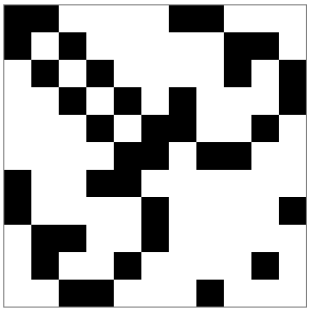
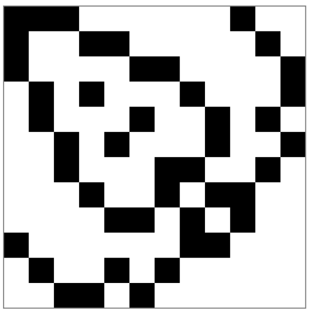
Each row in the tables in this section contains a value for a, a value for m and all maximal matrices for $z_a(m)$ up to isomorphism (which includes transposing the matrix) together with their row and column sums and automorphism groups. The matrices are presented both as images and as coded strings that can be decoded through the `decode_array()` function in Kyoto [15].

A $(0, 1)$-matrix is encoded by flattening it so that rows remain contiguous, padding the result on the right to a multiple of 8 bits with zeros, interpreting each byte in *little-endian* order and encoding the final byte sequence using Base64. The height and width are prepended, separated by spaces.

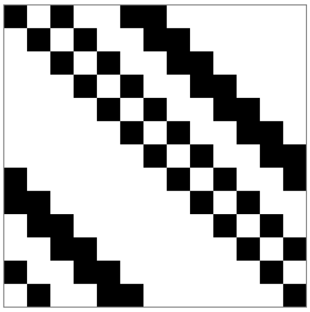
Where possible a symmetric presentation of each matrix has been chosen; those matrices without such representations have their encodings marked with an asterisk.

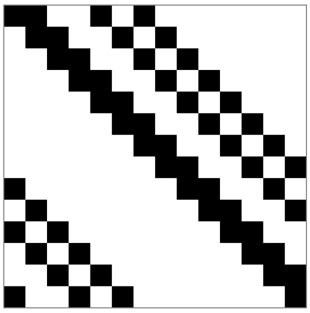
(a, m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
$(2, 2)$	3		`2 2 Bw==` $(2, 1)$ $(2, 1)$ C_2, order 2
$(2, 3)$	6		`3 3 qwE=` (2^3) (2^3) D_6, order 12

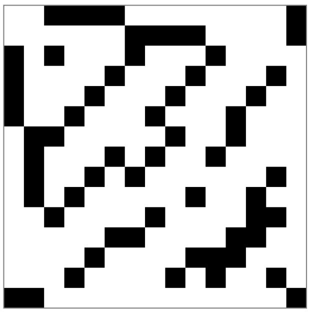
(a, m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
$(2, 4)$	9		`4 4 PpU=` $(3, 2^3)$ $(3, 2^3)$ $S_3 \times C_2$, order 12
$(2, 5)$	12		`5 5 zY6oAA==` $(3^2, 2^3)$ $(3^2, 2^3)$ C_2^2, order 4
			`5 5 fpQUAQ==` $(4, 2^4)$ $(4, 2^4)$ $S_4 \times C_2$, order 48
$(2, 6)$	16		`6 6 U6lyiQE=` $(3^4, 2^2)$ $(3^4, 2^2)$ D_8, order 16
$(2, 7)$	21		`7 7 CwsLGxMXAQ==` (3^7) (3^7) $\mathrm{PSL}(3, 2) \rtimes C_2$, order 336
$(2, 8)$	24		`8 8 CxYsWLBhwoU=` (3^8) (3^8) $\mathrm{GL}(2, 3) \rtimes C_2$, order 96
			`8 8 jhMlSWJUOIE=` $(4, 3^6, 2)$ $(4, 3^6, 2)$ D_6, order 12

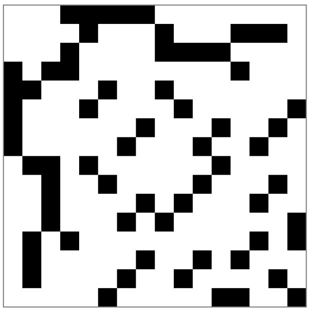
(a, m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
			`8 8 jhMlwWJUOAk=` $(4, 3^6, 2)$ $(4, 3^6, 2)$ C_2^2, order 4
$(2, 9)$	29		`9 9 PIgfSBIykiIxEgE=` $(4^2, 3^7)$ $(4^2, 3^7)$ D_{12}, order 24
$(2, 10)$	34		`10 10 cUhGKtIDFpSIESpwAA==` $(4^4, 3^6)$ $(4^4, 3^6)$ $S_4 \times C_2$, order 48
$(2, 11)$	39		`11 11 wyiYQqmIJthkIIQmkBAjAA==` $(4^6, 3^5)$ $(4^6, 3^5)$ D_{12}, order 24
$(2, 12)$	45		`12 12 B5JBYaiIIkWRxIQ0sBIYUsAC` $(4^9, 3^3)$ $(4^9, 3^3)$ $(C_3^2 \rtimes C_3) \rtimes D_4$, order 216

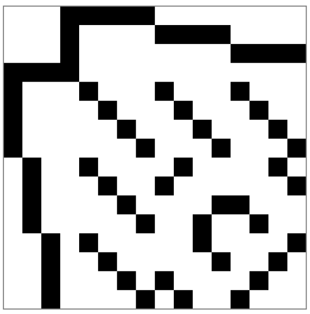
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group

$(2,13)$	52		`13 13 ZUAZUAaUAWVAG`
			`VAOlAPFQDHQDCQDAQ==`
			(4^{13}) (4^{13})
			$\mathrm{PSL}(3,3) \rtimes C_2$, order 11232

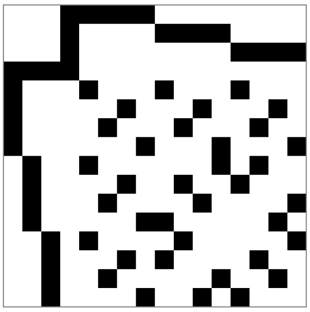
$(2,14)$	56		`14 14 U4ApwBRgCjAFmAJ`
			`MAaYBk4BZwChgFHAKCA==`
			(4^{14}) (4^{14})
			$(\mathrm{PSL}(3,2) \rtimes C_2) \times C_2$, order 672

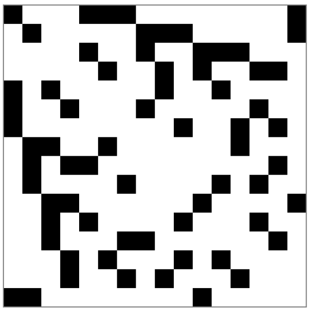
$(2,15)$	61		`15 15 PEDgYREhRBQRSUQYJ`
			`EQJUiAFCSEMDAPhQCgNAAE=`
			$(5^2, 4^{12}, 3)$ $(5^2, 4^{12}, 3)$
			$D_8 \times C_2$, order 32

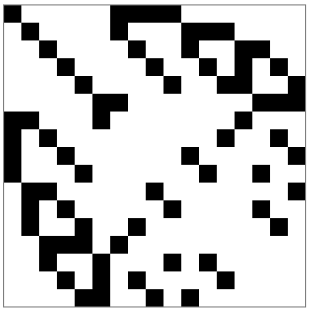
$(2,16)$	67		`16 16 +AAQcQgPDRAjARGCgUh`
			`BJBYIJESEIkSBCqCCFEJCIJg=`
			$(5^3, 4^{13})$ $(5^3, 4^{13})$
			D_6, order 12

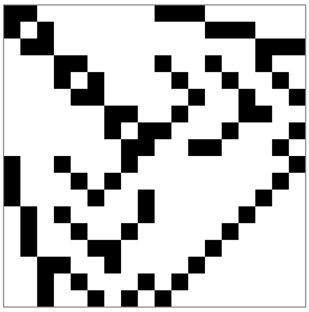
			`16 16 +AAIDwjwDwARESEiQUS`
			`BiBJCIoFCGIIkFIQkSEQhhBI=`
			$(5^3, 4^{13})$ $(5^3, 4^{13})$
			$S_4 \times S_3$, order 144

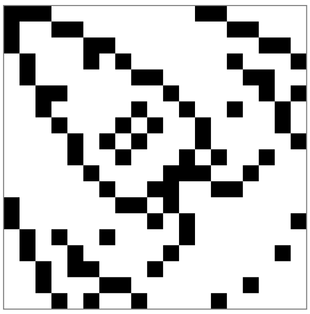
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group

			`16 16 +AAIDwjwDwAREUFEISK`
			`BiBIoQhIihIJBFIJEISRIhBQ=`
			$(5^3, 4^{13})$ $(5^3, 4^{13})$
			$S_3 \times D_4$, order 48

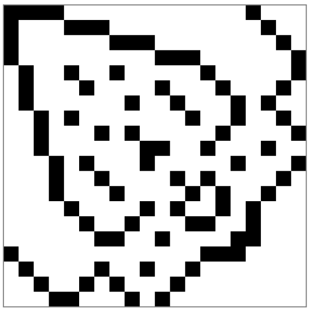
			`16 16 cYCCg5AcIGUFCYkgAVI`
			`mEBpAQigMhBQixEAoCkgRAwQ=`
			$(5^4, 4^{11}, 3)$ $(5^4, 4^{11}, 3)$
			C_2^3, order 8

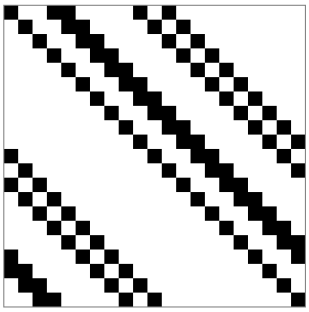
$(2,17)$	74		`17 17 wQOEOBCSQUgFIRMM+AiIAkg`
			`JBCOQGARUECIJiAsAiQJUCDAFAA==`
			$(5^6, 4^{11})$ $(5^6, 4^{11})$
			$C_2 \times S_3^2$, order 72

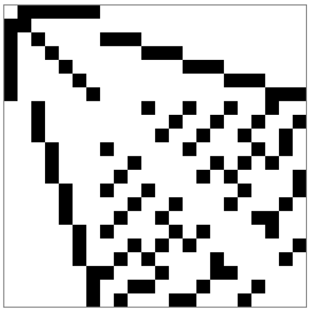
$(2,18)$	81		`18 18 Aw4UwGEAOIYkKCTBIAlMDNCIg`
			`BklAhgFUEggCkFIgiAGARMCFAWQCgA=`
			$(5^9, 4^9)$ $(5^9, 4^9)$
			$((C_3^2 \rtimes C_3) \rtimes C_2^2) \times C_2$, order 216

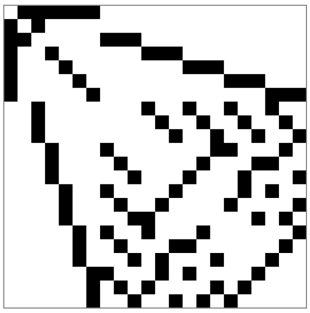
$(2,19)$	88		`19 19 BzDIAEYYwESBKDAYBoISJAlRQlARh`
			`EQJCCeAzBBYgAAFKiFAgkAOAiAGBEoIAA==`
			$(5^{12}, 4^7)$ $(5^{12}, 4^7)$
			$S_4 \times S_3$, order 144

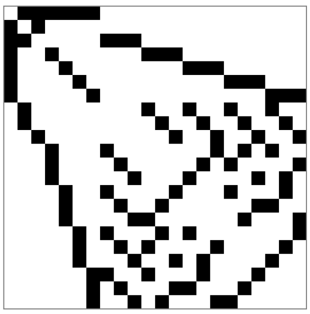
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group

$(2,20)$	96		`20 20 DwARByCBAxTAgZIgKE`
			`JEAolCAUlEQUhgIiiCiIRCiFA`
			`CoRQgMQFMGAHgICQBpAiAUQA=`
			$(5^{16},4^4)$ $(5^{16},4^4)$
			$((((C_2^4 \rtimes C_2) \rtimes C_2) \rtimes C_3^2) \rtimes C_2) \rtimes C_2$, order 2304

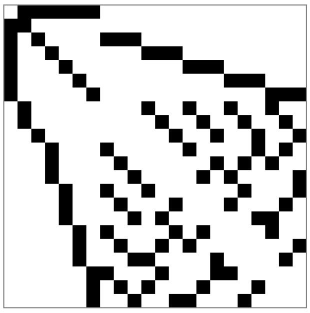
$(2,21)$	105		`21 21 GQpAhgKQoQBkKAAZCkCGA`
			`pChAGQoABkKQIYGkCEBZFgAGRRAB`
			`gWQQQFkUAA5FEAOBRBDAcRQAAE=`
			(5^{21}) (5^{21})
			$\mathrm{PSL}(3,4) \rtimes D_6$, order 241920

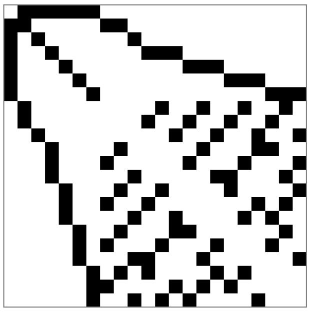
$(2,22)$	108		`22 22 fgDAAABQOAAkcAAR4EAIwBEE`
			`gBOQJASQJAGShAhCIQgqCEEhJIEIIR`
			`FBJDCgUAiICghSCAEjBkBGBFCMAA==`
			$(6,2,5^{20})$ $(6,2,5^{20})$
			$S_3 \times S_5$, order 720

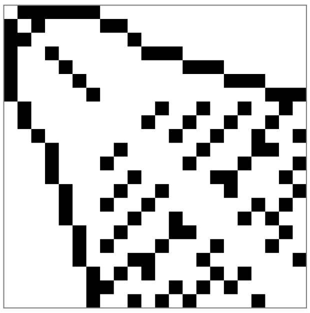
			`22 22 fgBAAQAwOAAkcAAR4EAIwBEE`
			`gBOQJARIEgEkiQgYIQQxCCIiJIQCkR`
			`BCGFCgRCBIDASiiACjEECFApBUAA==`
			$(6,2,5^{20})$ $(6,2,5^{20})$
			$S_3 \times S_4$, order 144

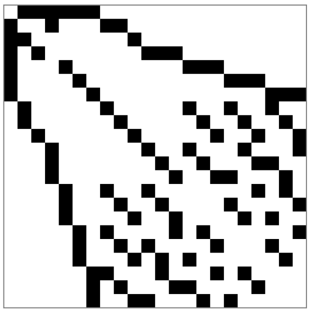
			`22 22 fgBAAQAwOAAkcAAR4EAIwBEE`
			`gAuQJAJIEgEkiQioIASFCCIUJEQEkc`
			`BAGIigKCBIIQQihQATEUAxApBiAA==*`
			$(6,2,5^{20})$ $(6,4,3,5^{19})$
			S_4, order 24

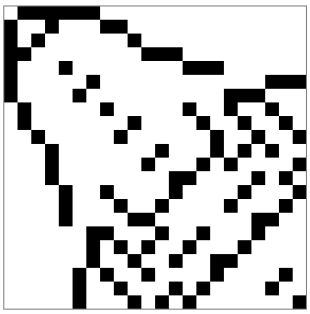
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group

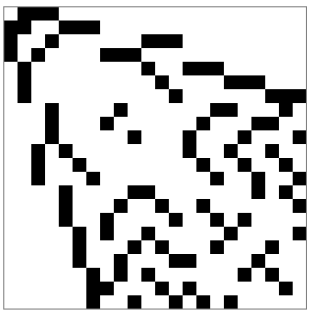
			`22 22 fgDAAABQOAAkcAAR4EAIwBEE`
			`gAuQJAJIEgEkiQhCIQQqCEIhJIEIER`
			`FBKDCgUAhICghiCAEjBkBFBJCMAA==*`
			$(6,2,5^{20})$ $(6,4,3,5^{19})$
			S_4, order 24

			`22 22 fgDAYABQIAAkcAAR4EAIwBEE`
			`gAsgSQIkCQEkiRDEIIKICIIRREIISU`
			`BBSCggMRAoIgJiBAIVCsBQAZAKAQ==`
			$(6,4,3,5^{19})$ $(6,4,3,5^{19})$
			$D_4 \times C_2$, order 16

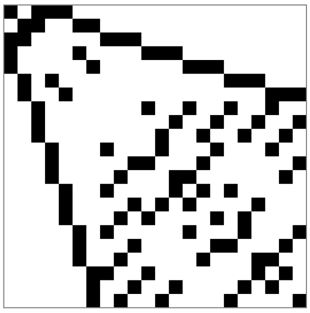
			`22 22 fgBAYQAwIAAkcAAR4EAIwBEE`
			`gAsgSQIkCQEkiRDEIIKICIIRREIISU`
			`BBSCggMRAoIgJiBAIVCsBQAZAKAQ==`
			$(6,4,3,5^{19})$ $(6,4,3,5^{19})$
			$D_4 \times C_2$, order 16

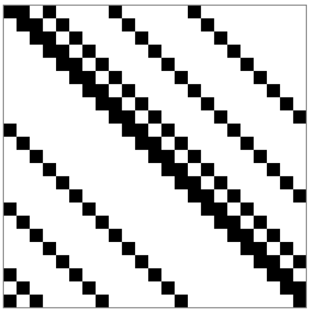
			`22 22 fgBAYgAwIAAUcAAR4EAIwBEE`
			`gAuCJAJBEoEgiUAiIiAxCJARJAEFkR`
			`BCSCigUCBIIQKiAgEjCkAxBJBRAA==*`
			$(6,4,3,5^{19})$ $(6,4,3,5^{19})$
			S_3, order 6

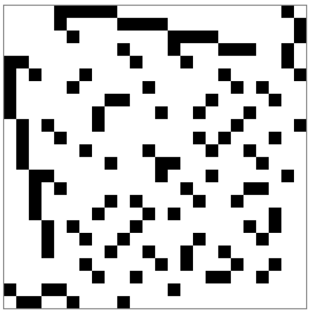
			`22 22 fgBAYgBQIAAMcAAR4EAQAB4C`
			`cAiCJAJCEkEgiYCoIBCFCDAUJIQIkR`
			`BBGDDASARQiQAkGYASQiBRCIgKCA==`
			$(6,4,3,5^{19})$ $(6,4,3,5^{19})$
			S_3, order 6

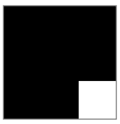
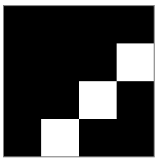
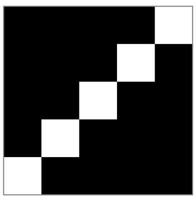
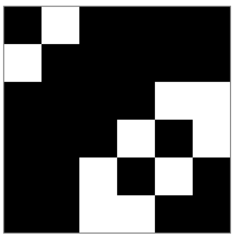
			`22 22 DgDAHACQwAEUDgAC5IAAwiEA`
			`gSMERohADIKISAGSkABJRIAkhAEFkQ`
			`RCQgqgBCGIIgISQwAVKMAoEJBUAA==`
			$(3,5^{21})$ $(3,5^{21})$
			$(C_3^2 \rtimes C_3) \rtimes C_2^2$, order 108

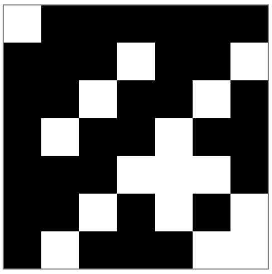
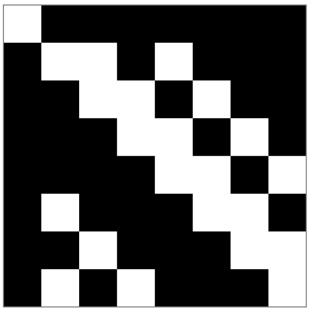
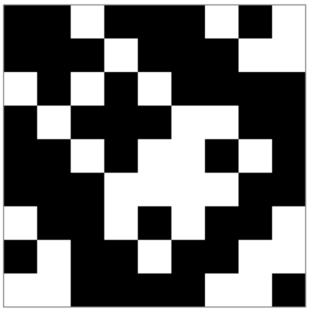
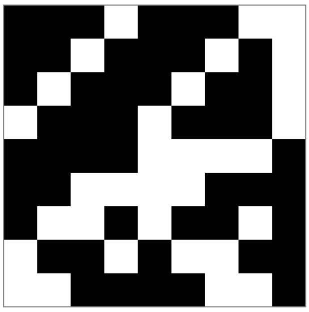
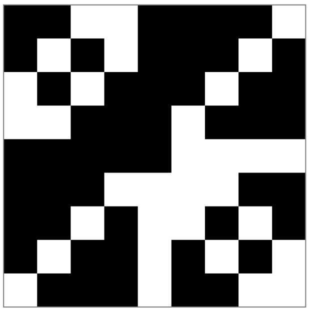
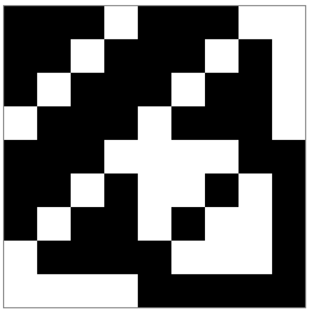
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group

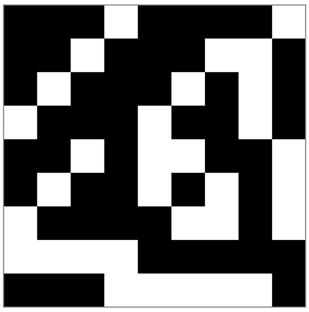
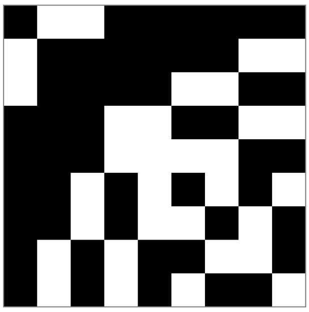
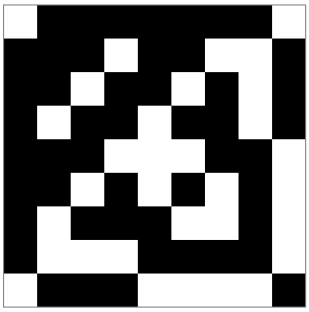
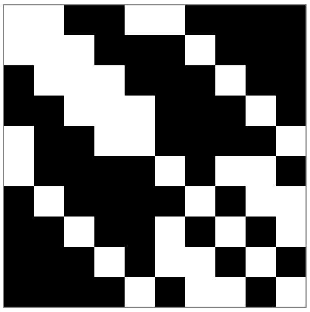
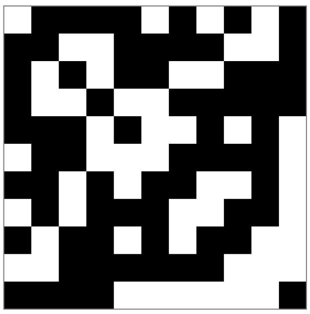
			`22 22 HQCAGQAwOACEcABB4IACwCEB`
			`gBOQJASQJAGShIiIIBiBCDEQJFQAoU`
			`JAFAqgICKIYAQSxAATUEASClBCCA==`
			$(4^2, 5^{20})$ $(4^2, 5^{20})$
			$D_4 \times S_3$, order 48

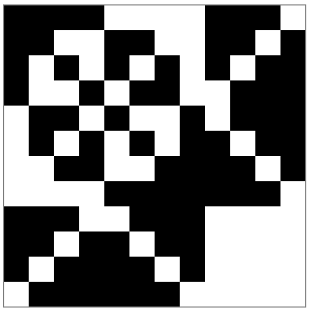
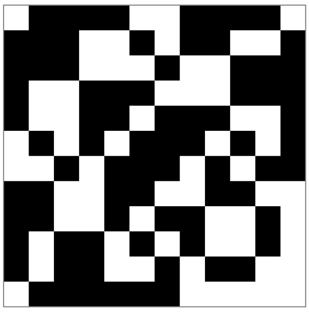
$(2,23)$	115		`23 23 C0EAC0EAC0EAC0EAC0EAC0EAC0`
			`EAC0EAC8EAC4EAC4EAC4EAC4EAC4EAC4`
			`MAC4IAC4IAC4IAC4IAG4IAE4IAF4IAAQ==`
			(5^{23}) (5^{23})
			D_{23}, order 46

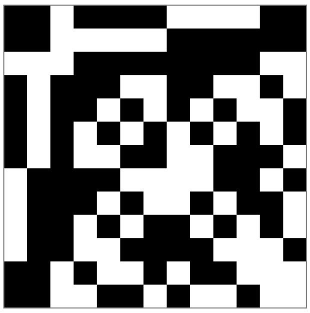
$(2,24)$	122		`24 24 8AFAEB6AIOCBACJOAORARQCCIQgU`
			`AQMhgZAIigCEEoAiQggJAjEQDBBBFEAYBI`
			`UEhCggKAQoSIIQCEkCQFAkgAQTGSAAJgIA`
			$(6^4, 5^{18}, 4^2)$ $(6^4, 5^{18}, 4^2)$
			D_8, order 16

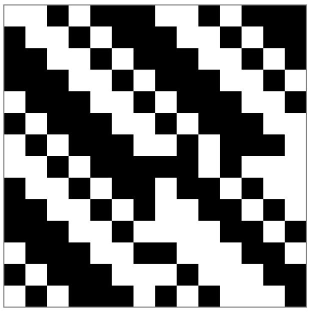
$(3,3)$	8		`3 3 /wA=` $(3^2, 2)$ $(3^2, 2)$ D_4, order 8
$(3,4)$	13		`4 4 f9s=` $(4, 3^3)$ $(4, 3^3)$ D_6, order 12
$(3,5)$	20		`5 5 7+7uAQ==` (4^5) (4^5) $S_5 \times C_2$, order 240
$(3,6)$	26		`6 6 vf9c6ww=` $(5^2, 4^4)$ $(5^2, 4^4)$ $D_8 \times C_2$, order 32

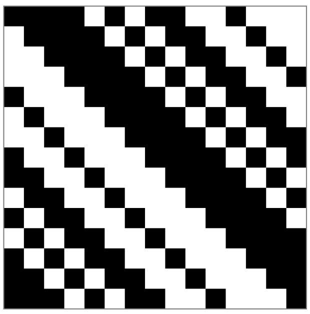
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
$(3,7)$	33		7 7 /tu2fVx1AA== $(6,5^3,4^3)$ $(6,5^3,4^3)$ D_6, order 12
$(3,8)$	42		8 8 /unTp0+dO3U= $(7,5^7)$ $(7,5^7)$ $\mathrm{PSL}(3,2) \rtimes C_2$, order 336
$(3,9)$	49		9 9 u+6o77z0sLU2PAE= $(6^4,5^5)$ $(6^4,5^5)$ C_2^2, order 4
			9 9 d3Z1c/dweFrLPAE=* $(6^4,5^5)$ $(6^4,5^5)$ D_4, order 8
			9 9 8+rq5v7h8NJWbgA= $(6^4,5^5)$ $(6^4,5^5)$ D_8, order 16
			9 9 d3Z1c3d4aUuP8AE= $(6^4,5^5)$ $(6^4,5^5)$ $S_4 \times C_2$, order 48

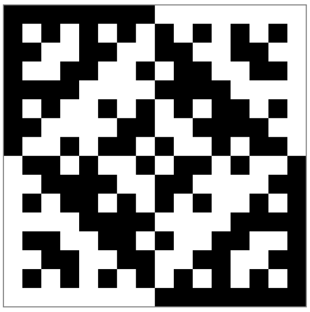
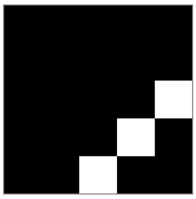
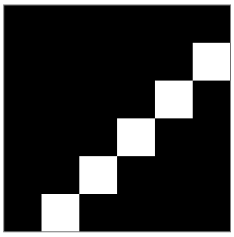
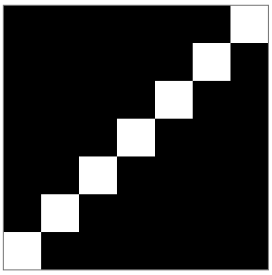
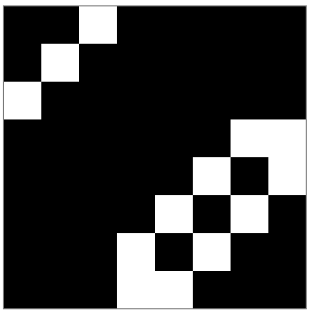
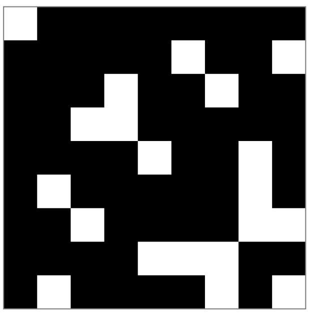
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
			`9 9 93Z2dbuslSf4BwE=*` $(7,6^3,5^4,4)$ $(6^4,5^5)$ S_3, order 6
			`9 9 +f14PnN41dKa1QA=` $(7,6^2,5^6)$ $(7,6^2,5^6)$ D_8, order 16
			`9 9 /m5ubXtsVad4DgE=` $(7,6^3,5^4,4)$ $(7,6^3,5^4,4)$ D_6, order 12
$(3,10)$	60		`10 10 zOMe97jmednLVpe+BA==` (6^{10}) (6^{10}) $S_5 \times C_2$, order 240
$(3,11)$	69		`11 11 Xp1nzZN/KeOtSWet4ccDAQ==` $(7^4,6^6,5)$ $(7^4,6^6,5)$ D_6, order 12

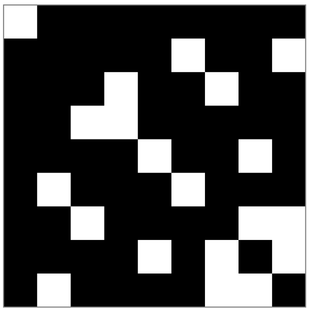
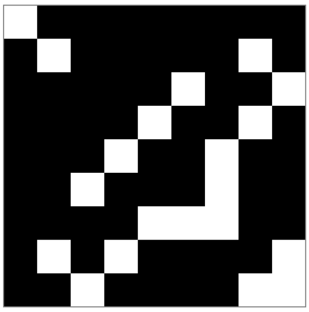
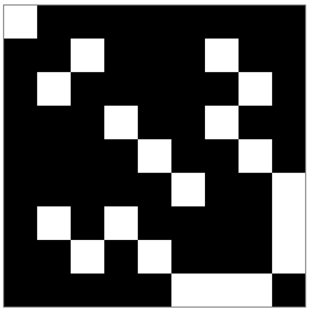
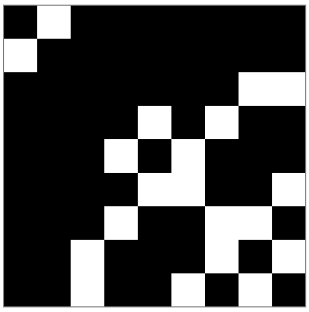
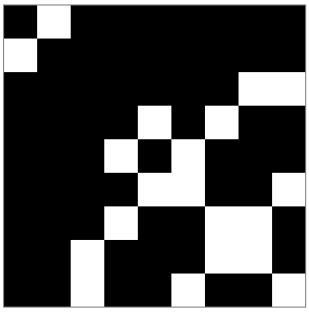
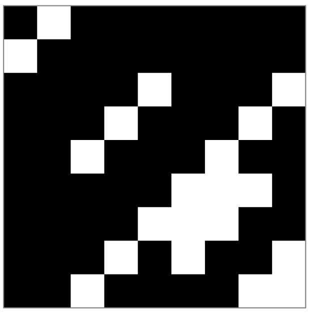
(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
$(3,12)$	80		`12 12 DzezVZ3mlq7azAt/57ANveAH` $(7^8,6^4)$ $(7^8,6^4)$ $(((C_2^3 \rtimes C_2^2) \rtimes C_3) \rtimes C_2) \rtimes C_2$, order 384
			`12 12 nneaR57j2amudD0z09RKTeMH` $(8,7^6,6^5)$ $(8,7^6,6^5)$ $D_4 \times C_2$, order 16

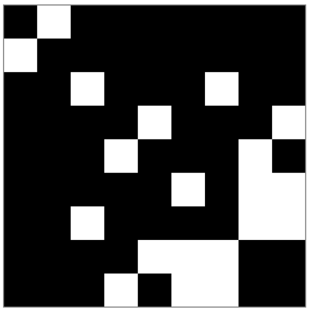
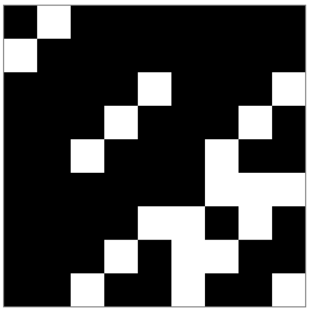
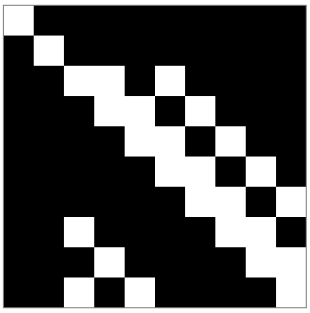
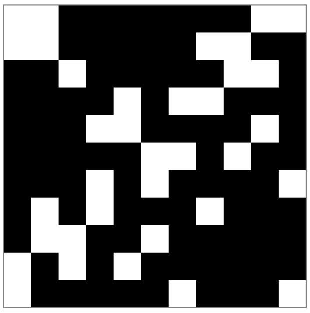
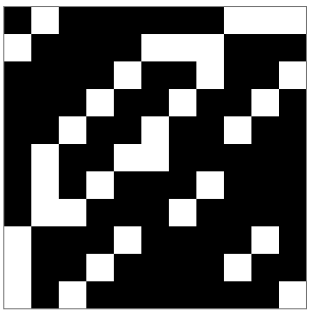
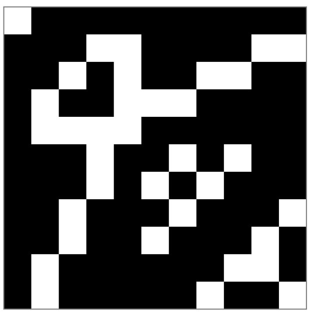
$(3,13)$	92		`13 13 e3jw45/O1Cqra`
			`pnzsC7NWpnHpTFLAA==`
			$(8^3,7^8,6^2)$ $(8^3,7^8,6^2)$
			$(C_8 \rtimes C_2^2) \rtimes C_2$, order 64

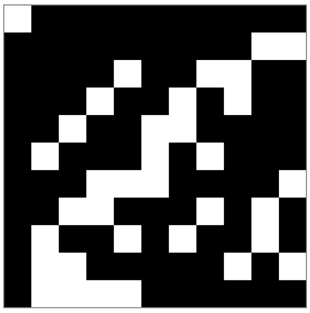
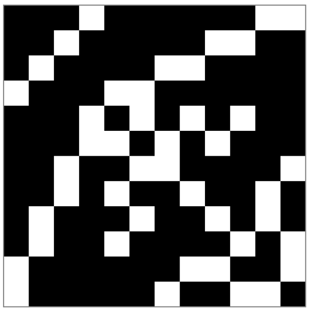
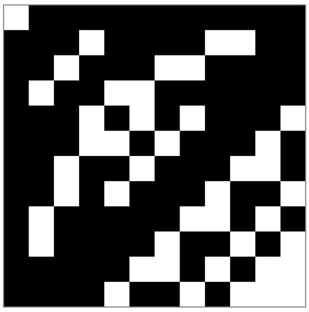
$(3,14)$	105		`14 14 dHo6PZ2eTk5np6P`
			`TORdpy5R1wjpjHbGuCA==`
			$(8^7,7^7)$ $(8^7,7^7)$
			$\mathrm{PSL}(3,2) \rtimes C_2$, order 336

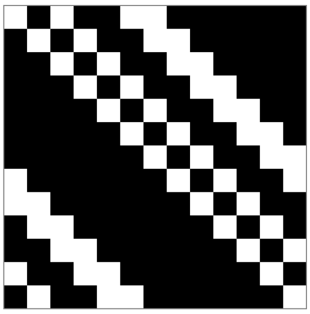
$(3,15)$	120		`15 15 rwmvCa8JrxmvEa8Rr`
			`xOvEy8TbxNPE18TXxNfEwE=`
			(8^{15}) (8^{15})
			S_8, order 40320

(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
$(3,16)^*$	128		`16 16 /wBVVTMzmWYPD6Vawzx paZaWPMNapfDwZpnMzKqqAP8=` (8^{16}) (8^{16}) $((C_2 \times (C_2^3 \rtimes C_2^2)) \rtimes C_2) \rtimes (\mathrm{PSL}(3,2) \rtimes C_2)$, order 43008
$(4,4)$	15		`4 4 /38=` $(4^3,3)$ $(4^3,3)$ $S_3^2 \rtimes C_2$, order 72
$(4,5)$	22		`5 5 /7+7AQ==` $(5^2,4^3)$ $(5^2,4^3)$ $D_4 \times S_3$, order 48
$(4,6)$	31		`6 6 //feew8=` $(6,5^5)$ $(6,5^5)$ $S_5 \times C_2$, order 240
$(4,7)$	42		`7 7 v+/7vu/7AQ==` (6^7) (6^7) $S_7 \times C_2$, order 10080
$(4,8)$	51		`8 8 +/3+P1+v1+c=` $(7^3,6^5)$ $(7^3,6^5)$ $C_2 \times D_5 \times S_3$, order 120
$(4,9)$	61		`9 9 /r/dnv+2757HvQA=` $(8,7^5,6^3)$ $(8,7^5,6^3)$ C_2, order 2

(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
			`9 9 /r/dnv+2+55XPQE=` $(8,7^5,6^3)$ $(8,7^5,6^3)$ C_2^2, order 4
			`9 9 /vt+e3t79+N6ewA=` $(8,7^5,6^3)$ $(8,7^5,6^3)$ C_2, order 2
			`9 9 /nf3vf32W711HwE=` $(8,7^5,6^3)$ $(8,7^5,6^3)$ D_4, order 8
			`9 9 /f3/eX392c1dWwE=` $(8^2,7^3,6^4)$ $(8^2,7^3,6^4)$ C_2^3, order 8
			`9 9 /f3/eX392c2d2wA=` $(8^2,7^3,6^4)$ $(8^2,7^3,6^4)$ C_2^3, order 8
			`9 9 /f2/u7v74+NrewA=` $(8^2,7^3,6^4)$ $(8^2,7^3,6^4)$ C_2^2, order 4

(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
			`9 9 /f3vfnf3y57HlwE=` $(8^2, 7^3, 6^4)$ $(8^2, 7^3, 6^4)$ C_2^3, order 8
			`9 9 /f2/u7v7x9PL2wA=` $(8^2, 7^3, 6^4)$ $(8^2, 7^3, 6^4)$ D_6, order 12
$(4, 10)$	74		`10 10 /vc//elPf/rTnvesBw==` $(9^2, 7^8)$ $(9^2, 7^8)$ $(\mathrm{GL}(2,3) \rtimes C_2) \times C_2$, order 192
$(4, 11)$	86		`11 11 /0HzPl9+3k9fr+7ZV7/vAA==` $(7^2, 8^9)$ $(7^2, 8^9)$ C_2^2, order 4
			`11 11 /fD422677ebXPffutbf+AA==` $(7^2, 8^9)$ $(7^2, 8^9)$ D_6, order 12
			`11 11 /j/Pmhsf/ltffXfb7WffAA==` $(10, 7^4, 8^6)$ $(10, 7^4, 8^6)$ D_{12}, order 24

(a,m)	$z_a(m)$	Maximal matrix	Code, row/column sums, automorphism group
			`11 11 /v/Pm2+9+a4fb66tzVf4AQ==*` $(10,9,8^4,7^5)$ $(10,7^4,8^6)$ D_4, order 8
			`12 12 97PPPe/8V33qm7e23dpefuab` $(9^4,8^8)$ $(9^4,8^8)$ $\mathrm{GL}(2,3) \rtimes C_2$, order 96
$(4,12)$	100		`12 12 /n/PO9/8V3e627lufdpbn/IW` $(11,9^3,8^6,7^2)$ $(11,9^3,8^6,7^2)$ D_{12}, order 24

$(4,13)$	117		`13 13 mr/mr/lr/pq/5`
			`q/xa/w6v84v89v8AA==`
			(9^{13}) (9^{13})
			$\mathrm{PSL}(3,3) \rtimes C_2$, order 11232

5.1 Discussion

Nearly all of the found maximal matrices can be arranged to be symmetric, with all exceptions at least having some other automorphisms. This suggests that exhaustively checking matrices for all-one minors as Collins [3] did is highly unlikely to yield maximal matrices, even if the tools used can support an exhaustive search at the desired matrix size. Instead, to obtain explicit lower bounds for the Zarankiewicz function, one should try extending smaller matrices to larger ones by adding as many ones as possible.

The first non-trivial maximal matrices for a given a follow a simple pattern: for $a \le m < 2a$ the complement of the bipartite graph equivalent to the maximal matrix is simply $2(m-a)+1$ isolated edges, and when $m = 2a$ the complement is $a-1$ isolated edges and a $2(a+1)$-cycle. Yang [19] has proved that these are indeed the unique maximal matrices up to isomorphism for these sets of parameters, but the situation immediately becomes very complicated after that point: there are 2 maximal matrices for $z_2(5)$, only 1 for $z_3(7)$, but 9 for $z_4(9)$.

Some maximal matrices in the above table have been arranged to highlight a circulant (sub)matrix motif. This is most apparent for the $z_2(m)$ cases solved by Reiman's projective plane construction [11], since the resulting matrix can always be made circulant by a result of Singer [13], but circulant matrices also appear elsewhere. For example, 23 is the smallest number n above 21 for which five elements of $\mathbb{Z}/n\mathbb{Z}$ can be chosen so that their pairwise differences are all distinct – $(0,1,3,8,14)$ is an example – and taking cyclic shifts of any such set yields, as it turns out, the unique maximal matrix for $z_2(23)$. (Even after enforcing the constraints in section 3 the CNF instance still has exactly 6^6 solutions, as counted by sharpSAT [16]; the matrix was verified to be unique by repeatedly shuffling and sorting its rows and columns, thereby reaching all 6^6 solutions.)

There is an asterisk in the row for $z_3(16)$ because even though its value has been shown to be exactly 128, the complete list of maximal matrices there has not yet been proven. The matrix shown is the only *known* maximal one up to isomorphism.

6 Conclusion

The CNF instances we generated for each set of parameters did not split the problem into cases finer than specific combinations of row and column partitions. Combined with the use of just one processor at a time, this imposed a limit on how far our new results could reach with reasonable computational effort at around $z = 100$. Parallelisation and further splitting outside the SAT solver (e.g. enumerating all possible ways to assign the first two rows and columns, each way leading to its own sub-case) as analysed in Heule [8] would help, but those techniques would in turn allow more optimisations which we did not consider either – argument D could exclude certain partial assignments without any further solving required, for example.

Despite the limitations, our results represent a significant contribution towards Zarankiewicz's problem, both in the range of new values and in revealing the structure of maximal matrices – previous work was mostly limited to the values, which by themselves follow no discernible pattern in general other than their strict monotonicity.

References

[1] Armin Biere et al. "CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling Entering the SAT Competition 2020". In: *Proceedings of SAT Competition 2020 – Solver and Benchmark Descriptions.* Ed. by Tomas Balyo et al. Vol. B-2020-1. Department of Computer Science Report Series B. University of Helsinki, 2020, pp. 51–53.

[2] Joshua Brakensiek et al. *The Resolution of Keller's Conjecture.* 2020. arXiv: `1910.03740 [math.CO]`.

[3] Alex F. Collins. "Bipartite Ramsey Numbers and Zarankiewicz Numbers". MA thesis. Rochester Institute of Technology, 2015.

[4] Karel Čulik. "Teilweise Lösung eines verallgemeinerten Problems von K. Zarankiewicz". In: *Annales Polonici Mathematici.* Vol. 1. 3. 1956, pp. 165–168.

[5] Paul Erdős and Arthur H. Stone. "On the structure of linear graphs". In: *Bulletin of the American Mathematical Society* 52.12 (1946), pp. 1087–1091. DOI: `10.1090/S0002-9904-1946-08715-7`.

[6] Richard K. Guy. "A many-facetted problem of Zarankiewicz". In: The Many Facets of Graph Theory. Springer, 1969, pp. 129–148. DOI: `10.1007/BFb0060112`.

[7] Tamás Héger. "Some graph theoretic aspects of finite geometries". PhD thesis. Eötvös Loránd University, 2013. URL: `https://heger.web.elte.hu/publ/HTdiss-e.pdf`.

[8] Marijn Heule. *Schur Number Five.* 2017. arXiv: `1711.08076 [cs.LO]`.

[9] Marijn Heule, Oliver Kullmann, and Armin Biere. "Cube-and-Conquer for Satisfiability". In: Handbook of Parallel Constraint Reasoning. Springer, 2018, pp. 31–59.

[10] Brendan D. McKay and Adolfo Piperno. "Practical Graph Isomorphism, II". In: *Journal of Symbolic Computation* 60 (2014), pp. 94–112. DOI: `10.1016/j.jsc.2013.09.003`.

[11] Istvan Reiman. "Über ein Problem von K. Zarankiewicz". In: *Acta Mathematica Academiae Scientiarum Hungarica* 9.3 (1958), pp. 269–273.

[12] Steven Roman. "A problem of Zarankiewicz". In: *Journal of Combinatorial Theory, Series A* 18.2 (1975), pp. 187–198. DOI: `10.1016/0097-3165(75)90007-2`.

[13] James Singer. "A theorem in finite projective geometry and some applications to number theory". In: *Transactions of the American Mathematical Society* 43.3 (1938), pp. 377–385.

[14] Carsten Sinz. "Towards an Optimal CNF Encoding of Boolean Cardinality Constraints". In: *International Conference on Principles and Practice of Constraint Programming.* Springer. 2005, pp. 827–831. DOI: `10.1007/11564751_73`.

[15] Jeremy Tan. *Exact values and maximal graphs for Zarankiewicz's problem.* URL: `https://github.com/Parcly-Taxel/Kyoto` (visited on 03/03/2022).

[16] Marc Thurley. *The #SAT solver sharpSAT.* URL: `https://github.com/marcthurley/sharpSAT` (visited on 03/03/2022).

[17] Nathan Wetzler, Marijn Heule, and Warren A. Hunt. “DRAT-trim: Efficient checking and trimming using expressive clausal proofs”. In: *International Conference on Theory and Applications of Satisfiability Testing*. Springer. 2014, pp. 422–429.

[18] Ed Wynn. *A comparison of encodings for cardinality constraints in a SAT solver*. 2018. arXiv: `1810.12975 [cs.LO]`.

[19] Chengcheng Yang. *A Problem of Erdős Concerning Lattice Cubes*. 2020. arXiv: `2011.15010 [math.CO]`.

[20] Kazimierz Zarankiewicz. “Problem P 101”. In: *Colloquium Mathematicae* 2.3–4 (1951), p. 301. URL: `https://eudml.org/doc/209990`.